Geostationary Operational Environmental Satellite (GOES)

GOES-R Series

GOES-R Reliable Data Delivery Protocol (GRDDP)

July 1, 2005

National Aeronautics and Space Administration

Goddard Space Flight Center
Greenbelt, Maryland

To verify the correct version of this document, please contact the GOES R Series Configuration Management Office.

Geostationary Operational Environmental Satellite (GOES)

GOES-R Series

GOES-R Reliable Data Delivery Protocol (GRDDP)

Prepared By:

6/30/2005

William Anderson, Muniz Engineering, Inc. Date
GOES-R Series Command and Data Handling Engineer

Approved By:

6/30/05

Alexander Krimchansky, NASA GSFC Date
GOES-R Series Systems Manager

7/1/05

Michael L. Donnelly, NASA GSFC Date
GOES-R Series Project Manager

GOES-R RELIABLE DATA DELIVERY PROTOCOL (GRDDP) CHANGE PAGE

TITLE: Geostationary Operational Environmental Satellites (GOES), GOES-R Series, GOES-R Reliable Data Delivery Protocol (GRDDP)

NUMBER: 417-R-RPT-0050

REVISION LTR/CHANGE NO.	EFFECTIVE DATE	DESCRIPTION OF CHANGES/PAGES AFFECTED
Baseline	December 1, 2004	Initial Release
Revision A	July 1, 2005	The title of the document was changed from "Geostationary Operational Environmental Satellite (GOES), GOES-R Series, GOES-R SpaceWire Transport Protocol" to "Geostationary Operational Environmental Satellite (GOES), GOES-R Series, GOES-R Reliable Data Delivery Protocol (GRDDP)" to clarify terminology commonly used in the Communications discipline. Revision A is a major rewrite of the Baseline document and changes affect all pages.

Table of Contents

Figures

Tables

1. INTRODUCTION

The GOES-R spacecraft uses European Cooperation for Space Standardization (ECSS) SpaceWire for the transfer of sensor, telemetry, and command data between instruments and the spacecraft. The GOES-R Program has directed that all data transferred over SpaceWire implement a reliable data delivery protocol. The SpaceWire Standard does not specify a protocol for reliable data delivery. It is the purpose of this document to specify a reliable data delivery protocol for the GOES-R spacecraft and instruments.

1.1 Scope

The Reliable Data Delivery Protocol uses the lower level SpaceWire data link layer to provide reliable packet delivery services to one or more higher level host application processes.

For GOES-R, the lower level protocol is the Packet Level service specified in the ECSS SpaceWire standard: ECSS-E-50-12A.

This document specifies the functional requirements for the Reliable Data Delivery Protocol service. This document does not specify the interfaces to the lower or higher level processes, which may be implementation dependent.

2. REFERENCE DOCUMENT

The lower layer protocol definitions for the GOES-R instrument to spacecraft data bus are compliant with EUROPEAN COOPERATION FOR SPACE STANDARDIZATION SpaceWire - Links, Nodes, Routers and Networks ECSS-E-50-12A, 24 January 2003.

3. DEFINITIONS

Transmitter
An electronic circuit that transmits signals over a physical medium.

Receiver
An electronic circuit that receives signals over a physical medium.

SpaceWire Port
SpaceWire transmitter and receiver circuits and associated logic that implements the SpaceWire Exchange level protocol including link initialization, character flow control, and link error detection and recovery.

SpaceWire Link
A bidirectional point-to-point connection between two SpaceWire ports.

Transport End Point
A Transport End Point (TEP) is defined on a host system for the purpose of either transmitting or receiving application packets over a SpaceWire Link. Multiple TEPs can be defined for any host system, but each TEP can only transmit or receive not both.

Transport Channel
A protocol defined data path between two TEPs. A Transport Channel can exist only between one transmit TEP and one receive TEP. Each Transport Channel is a one-way data path for application packets. The protocol supports multiple concurrent Transport Channels over a SpaceWire Link.

The Figure 1 illustrates the protocol objects and connections.

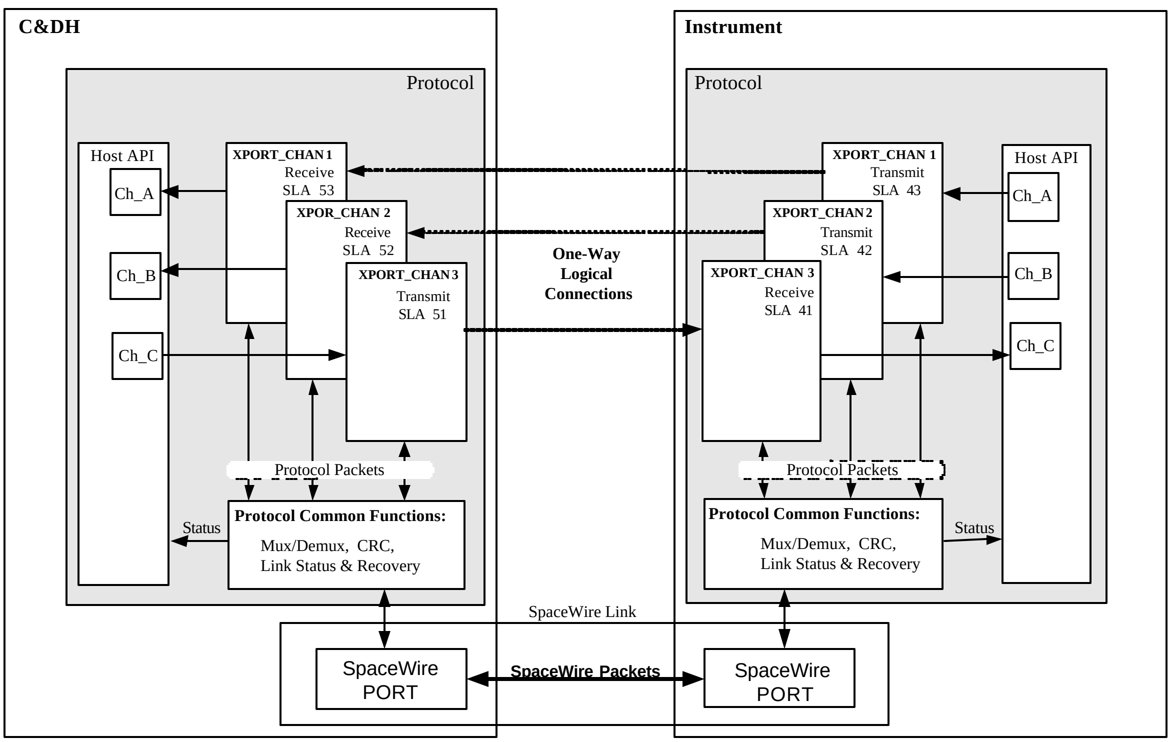


Figure 1. Protocol Objects and Connections

4. OVERALL FUNCTIONAL DESCRIPTION

The protocol transfers application data packets over a point-to-point SpaceWire connection that provides the SpaceWire services through the exchange and packet layers. The protocol adds the following capabilities to a SpaceWire link:

- Multiplexed Logical Connections
- Reliable Delivery

4.1 Multiplexed Logical Connections

The protocol shall support multiple simultaneous logical connections over a single SpaceWire link.

4.1.1 Channel Independence
Each logical channel shall operate independent of other logical channels.

4.1.2 Transmit Priority
When more than one packet is available for transmit, all Acknowledge packets shall be transmitted first, then Reset Command packets, then Data packets.

4.1.3 Transmit Queue
When data packets from more than one channel are available for transmit, packets shall be transmitted in the order in which they are queued.

4.2 Reliable Delivery

The transport protocol recovers from data that is damaged, lost, duplicated, or delivered out of order by the SpaceWire communication system. The protocol provides additional error detection beyond the SpaceWire parity, and utilizes packet sequence numbers, positive acknowledgement, and timeouts to detect lost or duplicated packets.

4.2.1 Error Detection
Packet errors shall be detected by adding a Cyclic Redundancy Check (CRC) to each packet transmitted, checking it at the receiver, and discarding any erroneous packet.

4.2.2 Packet Sequence Numbers
An 8 bit sequence number shall be assigned to each packet transmitted.

4.2.3 Sequence Number Use
At the receiver the sequence numbers shall be used to detect duplicates and to correctly order packets that may be received out of order.

4.2.4 Acknowledgement and Retransmit
The receiver shall send a positive acknowledgment (ACK) for each data packet received without error.

4.2.5 Retransmission
If the ACK is not received within a program defined timeout interval the data shall be retransmitted as defined in the Instrument to Spacecraft Interface Control Document.

5. PACKET FORMAT

All protocol packets include a 4-byte header, followed by a variable length payload, followed by a 1-byte CRC. Figure 2 shows how the protocol packet is encapsulated within the standard SpaceWire packet. Note that while a SpaceWire packet may have zero or more destination addresses before the payload, the Reliable Delivery Protocol requires that exactly one destination address is delivered to the protocol logic.

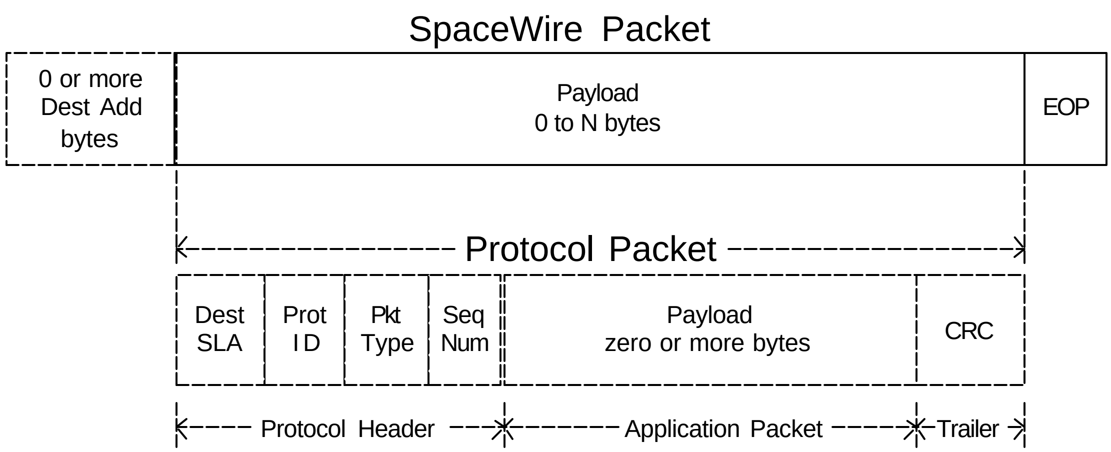


Figure 2. A GRDDP Packet within a SpaceWire Packet

5.1 Header

5.1.1 Destination Address

The first byte of the header shall be the Destination Address which is a one byte SpaceWire Logical Address (SLA). The SLA identifies the remote TEP to which the packet is being sent.

5.1.2 Protocol ID

The second byte of the header shall be the protocol ID assigned by the ECSS.

5.1.3 Packet Type

The third byte of the header shall identify the type of packet as listed in the Table 1.

Packet Type	Value
Application (Data) Packet	0
Acknowledge	1
Reset Command	2
(Reserved)	3…255

Table 1. Packet Type Values

5.1.4 Sequence Number

The fourth byte of the header shall be a sequence number in the range 0 through 255.

5.2 Payload

5.2.1 Data Packets
The protocol Data packets shall include a non-zero length payload that is the application packet.

5.2.2 Acknowledge and Reset Packets
Protocol packets that are an Acknowledge or Reset Command shall not include a payload.

5.3 Trailer

The protocol packet trailer shall be an 8 bit Asynchronous Transfer Code (ATM) Cyclic Redundancy Check (CRC) computed from the transport header destination SLA to the last payload byte, defined in the following polynomial:

CRC8, ATM (HEC)

$$\mathbf{x}^8 + \mathbf{x}^2 + \mathbf{x} + 1$$

6. TRANSPORT CHANNEL DEFINITION

The set of available Transport Channels for each host system shall be pre-defined in protocol configuration tables.

6.1 Transport Channel Parameters

Each Transport Channel shall be defined with the following parameters:

Local TEP Number	- A SpaceWire Logical Address (SLA) assigned to the Local TEP
Remote TEP Number	- The SLA of the TEP on the remote system to which the Local TEP is connected.
Channel Type	- From the local perspective, each channel is either a Transmit channel or a Receive channel.
Window Size	- The size of the channel's sequence number window.
Time Out	- Transmit channels only. The time to wait to receive an acknowledge before retransmitting a data packet.
Maximum Retries	- Transmit channels only.

6.2 TEP States

Each TEP shall be in one of three possible operating states:

(1) Closed – The TEP does not generate any packets on the link, and does not respond to any packets received.
(2) Enabled – A TEP transitions to the "Enabled" state when the host has requested it to be opened, and provided appropriate I/O buffer information. In addition, a Transmit TEP sends a reset command on this transition.
(3) Open – A Receive TEP transitions from Enabled to Open when a Reset command has been received from the remote Transmit channel.
A Transmit TEP transitions from Enabled to Open when it receives an ACK for a Reset command that is has sent to the remote Receive TEP.

7. CHANNEL OPERATIONS

7.1 Logical Connections

Upon power up initialization all TEPs shall be in the Closed state.

7.2 Reset Command

When a Transmit TEP transitions to the Enabled state, it shall send a Reset command to its remote channel and initiate an acknowledgement timer.

7.2.1 Reset Timer Cancellation

Upon receipt of a Reset acknowledgement, the transmit TEP shall cancel the acknowledgement timer.

7.2.2 Reset Timer Expiration

Upon expiration of the Reset timer period, the transmit TEP shall retransmit the Reset command.

7.3 Transport Channel Connection

A Transport TEP connection shall be considered established when a Transmit TEP and assigned Receive TEP are both in the Open state.

7.4 Receive TEP Operations

7.4.1 Receive TEP Data Packet

A receive TEP shall not send a data packet.

7.4.2 Receive TEP Reset Command

A receive TEP shall not send a reset command.

7.4.3 Receive Packet Order

A receive TEP shall use the data packet sequence numbers to deliver packets to the host in sequence number order.

7.4.4 Receive Duplicate Packets
A receive TEP shall use the data packet sequence numbers to discard duplicate packets.

7.4.5 Sliding Window
Each receive TEP shall maintain a sliding window which is a range of consecutive sequence numbers to determine whether each received data packet will be accepted or discarded.

7.4.6 Sliding Window Range
The receive window range shall start with the sequence number of the next packet expected to be delivered and end with sequence number equal to the start plus Window Size minus 1.

7.4.7 Window Advance
The receive window shall be advanced by 1 upon receipt of a packet containing the next expected sequence number.
Note: If packets with successively adjacent sequence numbers have already been received out of order, the start of the receive window will be advanced by more than 1, plus the number of successively adjacent "early" packets.

7.4.8 Packet Acknowledgement
All packets received without error shall be acknowledged.

7.4.9 Packets with Errors
Any packet received with detectable errors shall be discarded and not acknowledged.

7.4.10 Out of Window Sequence Number
A data packet that is received with a sequence number that is not within the receive window shall be acknowledged, but discarded.

7.4.11 Duplicate Sequence Number
A data packet received with a sequence number within the receive window that is a duplicate of a packet pending delivery to the host shall be acknowledged, but discarded.

7.4.12 Reset Command Sequence Number
A Reset command that does not have a sequence number of zero shall be treated as an error packet.

7.4.13 Reset Command Processing
When a Reset command is received, the receive window start shall be set to 1.

7.4.14 Packets Pending Delivery
All packets pending delivery to the host shall be discarded.

7.4.15 Reset Command Report
A reset command shall be reported to the host.

7.5 Transmit TEP Operations

7.5.1 Transmit TEP ACKs
A transmit TEP shall not send an ACK packet.

7.5.2 Transmit TEP Sequence Number Allocation
Each data packet transmitted shall have a sequence number allocated from the TEP's transmit window range of available sequence numbers.

7.5.3 Reset Command Sequence Number
All Reset command shall be transmitted with a sequence number zero.

7.5.4 Transmit Window
A transmit TEP shall maintain a sliding window range of consecutive sequence numbers that are available for transmitting data packets.

7.5.5 Unacknowledged Packets
The transmit window shall limit the number of unacknowledged data packets that can be transmitted.

7.5.6 Transmit Window Start
The transmit window start shall be set to 1 when an ACK is received for a Reset command.

7.5.7 Transmit Window Advance
The transmit window start shall be advanced by 1 when the ACK is received for the first sequence number in the transmit window.
Note: Similar to the case of advancing the Receive Window, out-of-order acknowledgements may require the Transmit Window to be advanced by more than 1 (1 plus the number of successively adjacent "early" acknowledgements).

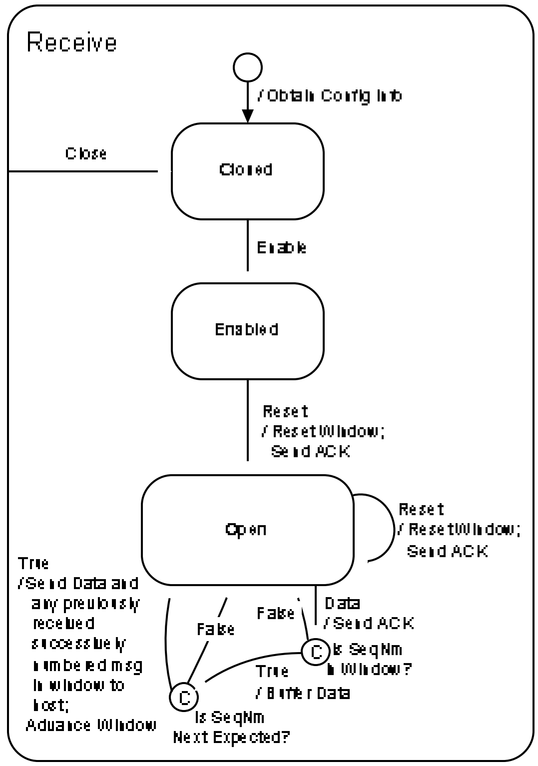
7.5.8 Packet Retransmit
A transmitted data packet that is not acknowledged within a program established timeout interval shall be retransmitted with the original sequence number up to Maximum_Retries number of times.

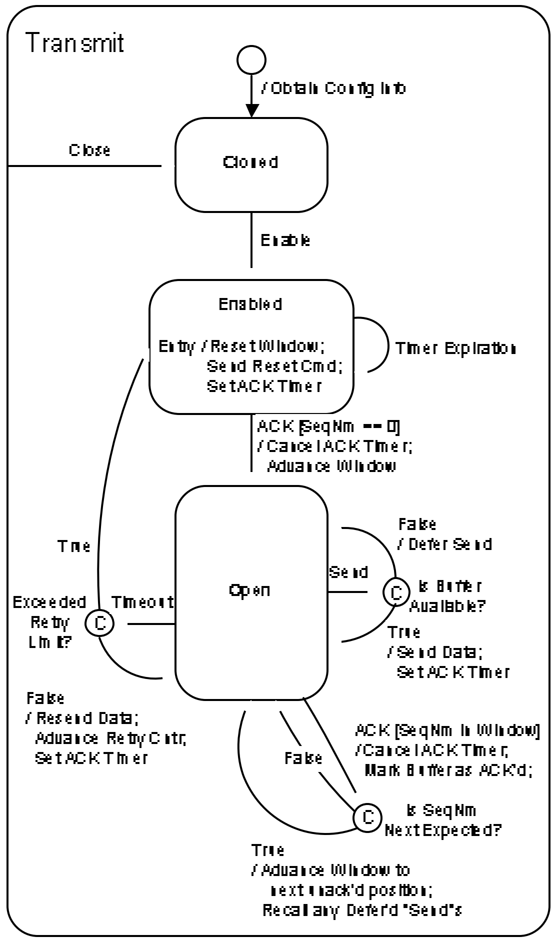
7.5.9 Timeout Start
The timeout interval shall begin when the last byte of the packet has been transmitted.

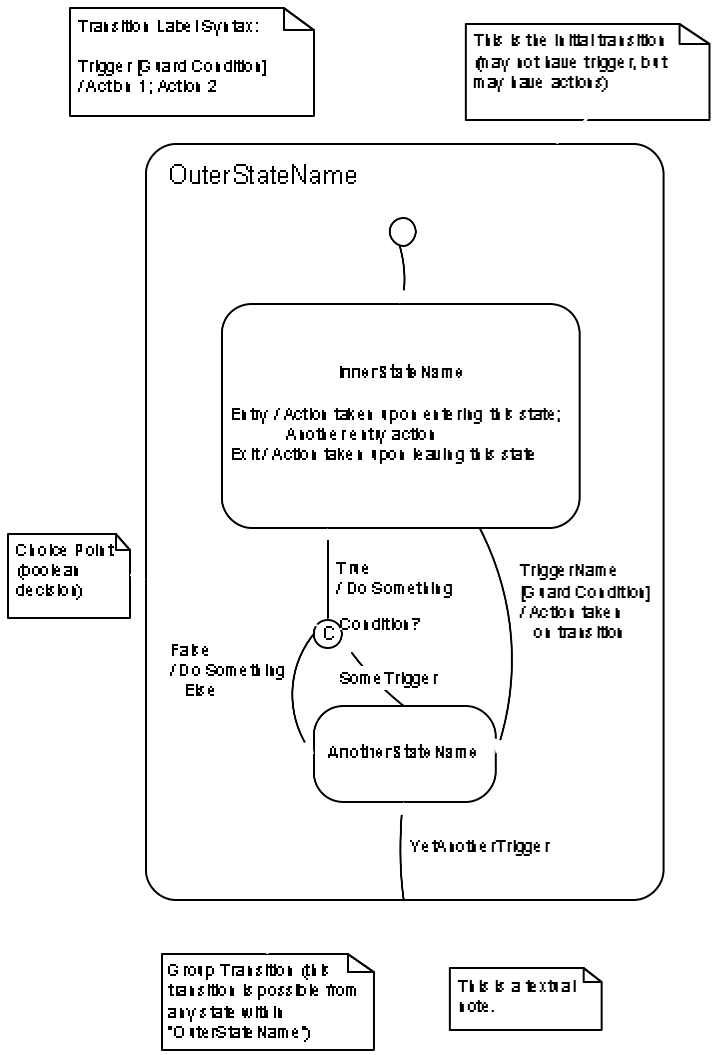
7.5.10 Timeout Report
When a transmitted Data packet has not been acknowledged after Maximum_Retries number of timeouts, the transmit channel shall report the error to the host and transition to the Enabled state.

7.5.11 Reset Command Retries
When in the Enabled state, a transmit channel shall retry the Reset command indefinitely until acknowledged.

APPENDIX A. STATE DIAGRAMS (Informative)




Transmit
/ Obtain Config Info
Close
Closed
Enable
Enabled
Entry / Reset Window;
Send Reset Cmd;
Set ACK Timer
Timer Expiration
ACK [SeqNm == 0]
/ Cancel ACK Timer;
Advance Window
True
Exceeded Retry Limit?
Timeout
Open
C
False
/ Resend Data;
Advance Retry Cntr;
Set ACK Timer
False
/ Defer Send
Send
C
Is Buffer Available?
True
/ Send Data;
Set ACK Timer
ACK [SeqNm in Window]
/Cancel ACK Timer;
Mark Buffer as ACK'd;
False
C
Is SeqNm Next Expected?
True
/ Advance Window to
next unack'd position;
Recall any Defer'd "Send"s


Transition Label Syntax:
Trigger [Guard Condition]
/Action 1; Action 2
This is the initial transition (may not have trigger, but may have actions)
OuterStateName
InnerStateName
Entry / Action taken upon entering this state;
Another entry action
Exit / Action taken upon leaving this state
Choice Point (boolean decision)
True
/ Do Something
C
Condition?
False
/ Do Something
Else
Some Trigger
TriggerName
[Guard Condition]
/ Action taken
on transition
AnotherStateName
YetAnotherTrigger
Group Transition (this transition is possible from any state within "OuterStateName")
This is a textual note.

APPENDIX B. ACRONYMS

ACK	Acknowledgment
ATM	Asynchronous Transfer Mode
CRC	Cyclic Redundancy Check
ECSS	European Cooperation for Space Standardization
GOES	Geostationary Operational Environmental Satellite
GOES-R	Geostationary Operational Environmental Satellite –R Series
GRDDP	GOES-R Reliable Data Delivery Protocol
GSFC	Goddard Space Flight Center
HEC	Header Error Code
ID	Identification
NASA	National Aeronautics and Space Administration
SLA	SpaceWire Logical Address
TEP	Transport End Point